Rushup

Document technique

# Architecture

L’architecture du projet est découpée en deux branches :

* InternalAssets : Nous rangeons ici les « assets » que nous créons nous même.
* StandardAssets : Nous rangeons ici les « assets » que nous récupérons depuis le « Unity Store ».

Nous rappelons qu’un « asset » est un élément (script, prefab, modèle, etc) ou groupe d’éléments utilisé dans la création du jeu.

# Récupération d’ « assets » existant (a.k.a ne pas réinventer la roue)

Afin de gagner du temps sur la réalisation du prototype, nous avons décidés d’utiliser plusieurs « assets » existants sur le store.

* FirstPersonCharacter : Il s’agit du personnage à la première personne que contrôle le joueur. L’asset possède les scripts de base permettant sa manipulation par le joueur.
* SimpleTurret : Il s’agit de la tourelle de base qui sera l’ennemie du joueur. Cet asset arrive avec les scripts de base permettant de faire tourner la tourelle, de viser le joueur et de lui tirer dessus.

# Adaptation des scripts existants et scripts additionnels

Afin de réalisé notre prototype nous avons ensuite adapté les asset existant et ajouté des scripts définissant les mécaniques que nous attendons.

Ces scripts se trouvent dans le dossier « InternalAssets/Scripts ».

## parkourFPSController

C’est le script gérant les mouvements du player :

* La course
* Le saut
* La course sur mur
* Le saut depuis le mur
* La glissade
* L’attaque

## LevelManager

## UIBehavior

Il s’agit du script définissant les informations affichées à l’écran

## TimeManager

## MusicHolderBehavior

## FadeInOut

## CheckpointBehavior

## CustomTurret

C’est ici que résident les scripts qui modifies ou étendes la tourelle de base.

### CustomShootingSystem

Ce script customise la manière que la tourelle de base a de tirer sur le joueur. En outre, il permet de faire tirer des obus à une fréquence réglable. Elle élimine également le reloading de la tourelle ou sa limite de munition.

# Problèmes rencontrés et améliorations possibles